Interspeech 2025
17-21 August 2025, Rotterdam, The Netherlands

Towards an Ultra-Low-Delay Neural Audio Coding with Computational Efficiency

Byeong Hyeon Kim[1], *Hyungseob Lim*[1], *Inseon Jang*[2], *Hong-Goo Kang*[1]

[1]Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea
[2]Electronics and Telecommunications Research Institute, Daejeon, South Korea

bhkim98@dsp.yonsei.ac.kr, hyungseob.lim@dsp.yonsei.ac.kr, jinsn@etri.re.kr, hgkang@yonsei.ac.kr

Abstract

Recent studies on neural audio codecs (NACs) have primarily focused on improving audio quality in extremely low bit-rate scenarios. However, they have not thoroughly explored the impact of latency. In this work, we first demonstrate that NACs can achieve high-quality reconstruction with an algorithmic delay below 1 ms, albeit at substantial computational costs. To address this challenge, we propose DualStream, a novel framework designed to significantly reduce computational costs in ultra-low-delay settings. DualStream integrates a lightweight encoding module with a small down-sampling ratio to maintain low algorithmic delay, combined with a larger module with a higher down-sampling ratio that processes time-delayed inputs to improve efficiency without introducing additional delay. Experimental results demonstrate that DualStream, with an algorithmic delay of 0.7 ms, achieves comparable performance to conventional NACs while reducing computational costs by approximately 40%.

Index Terms: neural audio coding, low-latency audio coding

1. Introduction

Audio coding (or audio compression) is essential for storing or transmitting digital audio signals. The primary goal is to minimize quality degradation after decoding while reducing the bit-rate required to represent the signal. In streaming applications, **latency**—the time elapsed between the start of processing at the sender and its arrival at the receiver—is a key performance metric for audio codecs. The acceptable latency threshold varies by application [1], ranging from up to 250 ms for telecommunications [2] to below 10 ms for live instrument monitoring [3].

Latency in audio communication can arise from various factors, including transmission delays, error concealment, and jitter buffering. However, minimizing the latency introduced by the audio coding algorithm itself, known as **algorithmic delay**, is critical. For example, transform coding, which converts time-domain audio signals into frequency-domain representations using filterbanks, introduces algorithmic delay due to the filter lengths or hop sizes used in forward and inverse transforms. To mitigate the algorithmic delay caused by the filter, legacy audio codecs have employed window functions with short overlaps between adjacent frames [4–6]. However, the fundamental trade-off between time and frequency resolution in transform coding poses a challenge, making it difficult for traditional codecs to achieve both low latency and high audio quality.

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [25ZC1100, The research of the basic media·contents technologies]

Recent advancements in deep learning have led to extensive research on neural network-based approaches for audio coding (neural audio codecs, or NACs). Conventionally, convolution neural network (CNN)-based autoencoder architectures combined with generative adversarial networks (GANs) have been widely adopted to achieve high-fidelity and realistic audio reconstruction [7–9]. Additionally, various frequency-domain training objectives have been explored to enhance perceptual quality. These include utilizing multiple short-time Fourier transform (STFT) time scales [10, 11], applying mel-filterbanks [12], and integrating psychoacoustic models [13] for perceptual optimization [14–16]. While most existing studies focus on improving audio quality at low bit-rates, relatively little attention has been given to reducing the algorithmic delay of NACs. In CNN-based NACs, the algorithmic delay is primarily determined by the sub-sampling ratio of convolutional layers, assuming all layers are designed to be causal. Typical full-band NACs, if implemented in a causal manner, exhibit algorithmic delays between 10 to 20 ms [7–9], which is significantly lower than the approximately 50 ms delay of traditional transform-based audio codecs [17]. However, given the existence of real-world use cases that require even lower latencies, we are motivated to study the potential for an ultra-low-delay NAC-based codec.

In this paper, we first establish a general NAC framework based on prior works [7–9] and evaluate the performance of a baseline model across different algorithmic delays. Our results demonstrate that NACs can preserve audio quality even when the algorithmic delay is reduced below 1 ms, albeit at the cost of increased computational complexity. To address this issue, we propose **DualStream**, a novel NAC architecture that enhances computational efficiency while maintaining high reconstruction quality, even at an extremely low algorithmic delay of less than 1 ms. The core idea behind DualStream is the utilization of two NACs, each with different down-sampling ratios to balance latency and computational efficiency. Specifically, we combine a small NAC with a low down-sampling ratio and a large NAC with a high down-sampling ratio. The larger NAC processes a time-delayed input signal, introducing no additional latency while significantly improving computational efficiency. This allows our method to achieve substantial computational savings compared to a conventional low-delay NAC of a similar model size. Experimental results show that that DualStream, with an algorithmic delay of 0.7 ms, reduces computational costs by over 40% while maintaining competitive audio quality.

2. Baseline NAC

We begin by constructing a baseline NAC using a framework similar to typical GAN-based NACs [7–9], which allows

10.21437/Interspeech.2025-1369

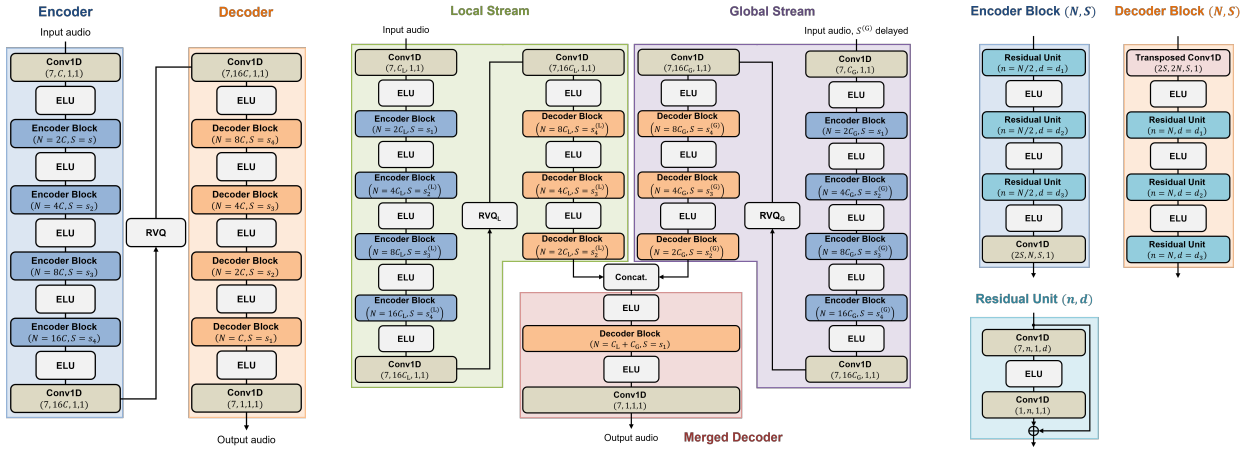


Figure 1: *(a) Architecture of the baseline NAC. (b) Architecture of DualStream. (c) Building blocks shared between the baseline NAC and DualStream. In the figure, the four arguments of the Conv1D layers denote the kernel size, output channel dimension, stride factor, and dilation factor, respectively.*

us to evaluate coding performance across various latency settings. The baseline NAC consists of three main components—encoder, quantizer, and decoder—as shown in Figure 1(a).

2.1. Model Architecture

The encoder processes the input signal $\boldsymbol{x} \in \mathbb{R}^T$ through an initial one-dimensional convolutional layer (Conv1D), followed by four Encoder Blocks (as shown in Figure 1(c)), and concludes with a final Conv1D layer, where T represents the length of the input audio signal. The channel dimension of the input signal is initially expanded to C after the first Conv1D layer and is subsequently doubled after each Encoder Block. Each Encoder Block consists of three consecutive Residual Units (as shown in Figure 1(c)) and a strided Conv1D layer. Given the stride factors $(s_1, \cdots, s_4)$ for strided Conv1D layers, the input audio signal x is down-sampled along the time axis by a factor of $S = \prod_{i=1}^{4} s_i$ during the processing of the encoder. Consequently, the encoder outputs a latent embedding sequence $\boldsymbol{y} = \left(\boldsymbol{y}_0, \cdots, \boldsymbol{y}_{(T/S)-1}\right) \in \mathbb{R}^{16C \times (T/S)}$. For the quantizer, we use residual vector quantization (RVQ) [18], building upon previous works [9, 19]. Given the sampling rate of the input audio as f_s, RVQ with N_vq VQ layers, each with N_cb vectors in its codebook, has a maximum bit-rate of $(f_\mathrm{s}/S) \times N_\mathrm{vq} \times \lceil \log_2 N_\mathrm{cb} \rceil$ (bps).

The decoder receives the quantized latent embeddings $\hat{\boldsymbol{y}}$ and generates the reconstructed signal $\hat{\boldsymbol{x}} \in \mathbb{R}^T$. The decoder follows a similar architecture to the encoder, but with the layers arranged in reversed order, and the down-sampling layers are replaced with transposed Conv1D layers [20]. All Conv1D layers in the baseline NAC are designed to be causal, ensuring that the reconstruction depends only on the current and past samples, and not on future samples. Additionally, an ELU activation function [21] is applied between each Conv1D layer, and spectral normalization [22] is applied to all Conv1D layers to enhance training stability [23].

2.2. Algorithmic Delay

Figure 2(a) illustrates how the algorithmic delay of the baseline NAC is calculated. Since the encoder consists of causal Conv1D layers with an overall down-sampling ratio of S, the

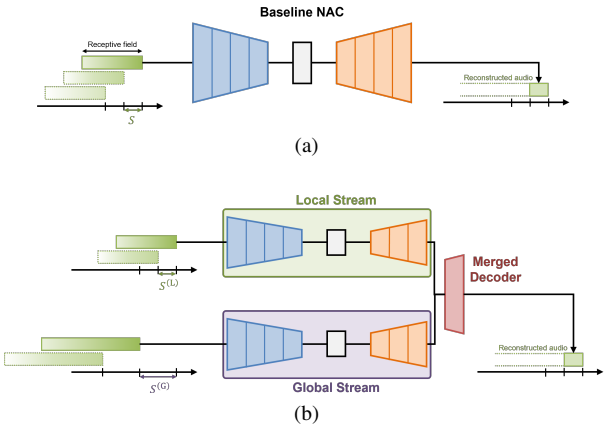


Figure 2: *Architectural delay of (a) the baseline NAC, and (b) DualStream.*

encoder must wait for S input audio samples before producing a quantized embedding vector. Consequently, the decoder can reconstruct the corresponding S samples only after the embedding vector has been transmitted. Thus, the algorithmic delay of the NAC is $S - 1$. It is important to note that the algorithmic delay is influenced solely by the down-sampling ratio S, not by the kernel size or receptive field, due to the causality.

3. DualStream

Experiments with baseline NACs, as discussed in Section 4, indicate that NACs with lower architectural delay can achieve audio quality comparable to that of NACs with higher delay, though at the cost of increased computational demands. Motivated by these observations, we propose a novel NAC architecture, **DualStream**, designed to achieve computational efficiency in ultra-low-delay configurations.

3.1. Model Architecture

As shown in Figure 1(b), DualStream is composed of a **Global Stream**, which has a high down-sampling ratio $S^{(\mathrm{G})} = \prod_{i=1}^{4} s_i^{(\mathrm{G})}$ but introduces no algorithmic delay; a **Local Stream**

Table 1: *Configuration of the baseline NACs and DualStreams.*

Model	Stream	s_i	C	N_{vq}, N_{cb}	Delay (ms)	# Params (M)	GMACs
Base_{512}		$(2, 4, 8, 8)$	48	16, 1024	11.59	30.16	46.36
Base_{256}		$(2, 4, 4, 8)$	48	8, 1024	5.78	28.80	52.36
Base_{128}		$(2, 4, 4, 4)$	48	4, 1024	2.88	24.00	50.53
Base_{64}		$(2, 2, 4, 4)$	48	2, 1024	1.43	23.78	79.39
Base_{32}		$(2, 2, 2, 4)$	48	1, 1024	0.70	23.20	109.45
$\text{Dual}^{a:b}_{128,32}$	Global	$(2, 4, 4, 4)$	40	-	0.70	22.50	65.25
	Local	$(2, 2, 2, 4)$	24	-			

Table 2: *Configuration of RVQs from* $\text{Dual}^{a:b}_{128,32}$ *at different bit-rate allocation ratios.*

	$a:b$	5:5	3:7	2:8	1:9	0:10
Global Stream	N_{vq}	2	2	1	1	-
	N_{cb}	1024	64	256	16	-
	Bit-rate (kbps)	6.89	4.13	2.76	1.38	-
Local Stream	N_{vq}	1	1	1	1	1
	N_{cb}	32	128	256	512	1024
	Bit-rate (kbps)	6.89	9.65	11.03	12.40	13.78

with a low down-sampling ratio $S^{(\text{L})} = \prod_{i=1}^{4} s_i^{(\text{L})}$, resulting in low algorithmic delay; and a **Merged Decoder** that takes as input the concatenated outputs from the two streams and produces the reconstructed audio signal $\hat{\boldsymbol{x}}$. $s_i^{(\text{G})}$ and $s_i^{(\text{L})}$ denote the stride factors of the i-th Encoder Block in the Global and Local Streams, respectively, where $s_1^{(\text{G})} = s_1^{(\text{L})} = s_1$ must be satisfied. Both streams follow the architecture of the baseline NAC, with the exception of the last Decoder block and Conv1D layer.

The Local Stream takes time-aligned audio signals $\boldsymbol{x} = (x_0, \ldots, x_{T-1}) \in \mathbb{R}^T$ as input and produces a feature map $\boldsymbol{h}_{\text{L}} \in \mathbb{R}^{2C_{\text{L}} \times (T/s_1)}$, where C_{L} denotes the channel size of the Local Stream. In parallel, the Global Stream processes a time-delayed audio signal by $S^{(\text{G})}$ samples, formulated as $\boldsymbol{x}_{\text{delay}} = (0, \ldots, 0, x_0, \ldots, x_{T-S^{(\text{G})}-1}) \in \mathbb{R}^T$, and outputs $\boldsymbol{h}_{\text{G}} \in \mathbb{R}^{2C_{\text{G}} \times (T/s_1)}$, where C_{G} is the channel size. In other words, the Global Stream operates on time steps that have already been processed by the Local Stream (Figure 2(b)). As a result, the Global Stream introduces no additional latency despite its large down-sampling ratio, while only the Local Stream contributes an algorithmic delay of $S^{(\text{L})} - 1$ samples. Finally, the output feature maps from both streams are concatenated along the channel dimension as $\boldsymbol{h}_{\text{M}} = (\boldsymbol{h}_{\text{L}}, \boldsymbol{h}_{\text{G}}) \in \mathbb{R}^{2(C_{\text{L}}+C_{\text{G}}) \times (T/s_1)}$ and fed into the Merged Decoder, which consists of one Decoder Block with channel size $(C_{\text{L}} + C_{\text{G}})$ and a final Conv1D layer, to produce a time-aligned reconstructed audio $\hat{\boldsymbol{x}} = (\hat{x}_0, \ldots, \hat{x}_{T-1}) \in \mathbb{R}^T$.

Assume a baseline NAC with stride factors $(s_1^{(\text{L})}, \ldots, s_4^{(\text{L})})$ and the same number of parameters as DualStream. Then, with a sufficiently large C_{G}, DualStream requires less computation than the baseline NAC due to the computational efficiency of the Global Stream with its large down-sampling ratio.

3.2. Stream-wise Training

Since DualStream consists of two autoencoders with different configurations merged at the last few layers, directly training the entire model from scratch may lead to suboptimal performance. Specifically, we often observed that one of the two streams produced zero-vectors after training from scratch. To prevent this issue, we apply a pre-training scheme called **Stream-wise Training** to ensure that both streams contribute to the reconstruction. During the Stream-wise Training phase, each mini-batch is divided into three groups. In the first group, the Local Stream is detached, and the corresponding output feature map $\boldsymbol{h}_{\text{L}}$ is replaced with a zero-vector. Similarly, the Global Stream is deactivated in the second group. In the final group, the entire DualStream is used for training.

4. Experiments and Results

4.1. System Configurations

Table 1 presents the detailed configuration of both the baseline NACs and DualStream. We begin by constructing five baseline NACs, each varying only in stride factors and quantizers, to analyze the effect of algorithmic delay on performance. DualStream variants are configured to have similar model sizes to Base_{32} and an algorithmic delay of 0.70 ms, with the Local Streams having a ratio of 32. The channel sizes of the streams, denoted as C_{G} and C_{L}, are set such that the Global Stream is approximately three times larger than the Local Stream. All models use dilation factors of $(d_1, d_2, d_3) = (1, 3, 9)$. The quantizers for both the baseline NACs and DualStream models are configured to have a maximum bit-rate of 13.78 kbps, where $(a : b)$ in Table 2 indicates that the maximum bit-rates of the Global and Local Streams have a ratio of a to b.

4.2. Experimental Setups

The NAC training objective consists of various loss functions: multi-scale mel-spectrogram reconstruction loss ($\mathcal{L}_{\text{Mel}}$) [9]; hinge adversarial loss ($\mathcal{L}_{\text{Adv_G}}$) [24] and feature matching loss ($\mathcal{L}_{\text{FM}}$) for adversarial training; and codebook loss ($\mathcal{L}_{\text{CB}}$) and commitment loss ($\mathcal{L}_{\text{CM}}$) [25] for joint training of the RVQ:

$$\mathcal{L}_{\text{Mel}} = \sum_{i=1}^{7} \| \log_{10} M_{2^{4+i},5\cdot 2^i} - \log_{10} \widehat{M}_{2^{4+i},5\cdot 2^i} \|_1, \quad (1)$$

where $M_{w,m}$ and $\widehat{M}_{w,m}$ represent the mel-spectrograms of the input and reconstructed audio signals, respectively. They are computed using a window length of w, a hop length of $w/4$, and m mel bins. We use three band-split STFT discriminators [9] and five multi-period discriminators [12] for adversarial training, following the configuration of DAC [9] and Hifi-GAN [12].

We apply a loss balancer [8] on $\mathcal{L}_{\text{Mel}}$, $\mathcal{L}_{\text{Adv_G}}$, and $\mathcal{L}_{\text{FM}}$ with weights of $\lambda_{\text{Mel}} = 10$, and $\lambda_{\text{FM}} = \lambda_{\text{Adv_G}} = 3$ to stabilize training. The hinge loss for the discriminators ($\mathcal{L}_{\text{Adv_D}}$), and loss for the quantizer ($\mathcal{L}_{\text{CB}}$ and $\mathcal{L}_{\text{CM}}$) are back-propagated without the balancer, using weights of $\lambda_{\text{Adv_D}} = 30$, $\lambda_{\text{CB}} = 4$, and $\lambda_{\text{CM}} = 1$. We configure the Adam optimizer with $\beta_1 = 0.8, \beta_2 = 0.99$, a learning rate of 1.0×10^{-4}, and a batch size of 8. We apply Stream-wise Training for the first 20,000 iterations, where either the Global or Local Stream is deactivated in 3 mini-batches, and remaining 2 mini-batches are trained with the entire DualStream. For training and validation, we select subsets of clean speech data[1] from the DNS Challange 4 dataset [26]. We use MUSDB18-HQ [27] and mtg-jamendo [28] for music data and FSD50K [29] for general audio. All the signals are converted to mono channel and resampled to a sampling rate of 44.1 kHz.

4.3. Performance Evaluation

For objective evaluation metrics, we used ViSQOL [30], noise-to-mask ratio (NMR) extracted from PEAQ [31], and 2f-model score [32]. NMR quantifies the amount of audible reconstruction errors, while ViSQOL and 2f-model estimate subjective evaluation scores. For the evaluation data, we selected clean

[1]We used emotional_speech, read_speech, VocalSet_48kHz_mono, and vctk_wav48_silence_trimmed from the clean_fullband directory.

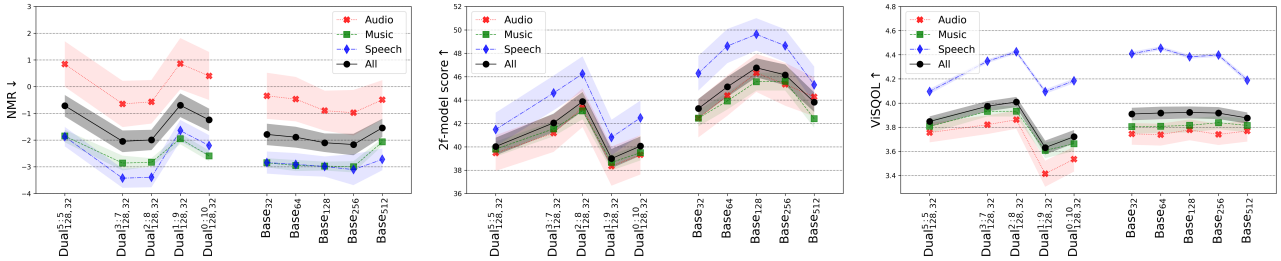


Figure 3: *Objective evaluation results of baseline NACs and DualStreams at 13.78 kbps.*

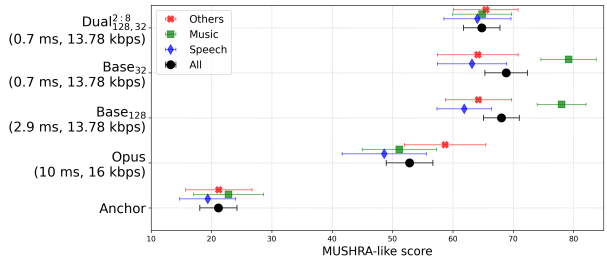


Figure 4: *Subjective evaluation results. Values in the parentheses indicate the algorithmic delay and the bit-rate.*

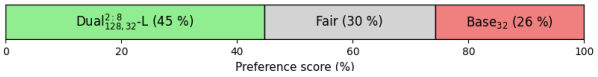


Figure 5: *AB preference test results between* Base_{32} *and* $\mathrm{Dual}^{2:8}_{128,32}$-L *with similar computational costs.*

speech from the DAPS dataset [33], music from the MUSDB18-HQ test data, and audio from the FSD50K evaluation data, each with a total length of approximately 30 minutes.

Figure 3 demonstrates the objective evaluation results with 95% confidence intervals. Among the five baseline NACs, Base_{128} yeilds the best 2f-model score, while all baseline NACs are competitive on NMR and ViSQOL scores. In summary, the baseline NAC maintains comparable performance across a wide range of delays. The NMR and 2f-model scores from DualStreams improve as more bits are allocated to the Local Stream, except for $\mathrm{Dual}^{1:9}_{128:32}$ and $\mathrm{Dual}^{0:10}_{128:32}$. $\mathrm{Dual}^{1:9}_{128:32}$ collapsed during training, even with Stream-wise Training, resulting in sub-optimal performance. Compared to the baseline, the best-performing DualStream, $\mathrm{Dual}^{2:8}_{128:32}$, achieves performance comparable to Base_{32}, although it slightly underperforms some other baseline NACs with higher algorithmic delay.

We also conducted MUSHRA (MUltiple Stimulus with Hidden Reference and Anchors) [34]-like subjective evaluations with 12 participants, which included $\mathrm{Dual}^{2:8}_{128:32}$, Base_{32}, Base_{128}, Opus [6] at 16 kbps with a frame duration of 10ms, reference signals, and the reference signals low-pass-filtered with a cutoff frequency of 3.5 kHz as anchors. The test consisted of 4 clean speech samples, 4 music samples, and 4 mixture samples of speech, music, environmental audio, or noise. Figure 4 shows the results with 95% confidence intervals, excluding the reference. All the neural-network-based codecs outperform the signal-processing-based Opus, even with a larger delay. This also verifies the effectiveness of neural networks for ultra-low-delay audio coding tasks. $\mathrm{Dual}^{2:8}_{128:32}$ demonstrates slightly lower scores on the entire test set than the baseline NACs, but still within the confidence interval. Overall, $\mathrm{Dual}^{2:8}_{128:32}$ demonstrates comparable performance to the baselines on speech and

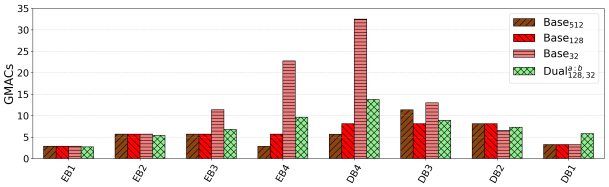


Figure 6: *Comparison of computational cost of each Encoder Block and Decoder Blocks, in terms of MACs. EB and DB denote the Encoder Block and Decoder Block, respectively.*

mixture signals, while exhibiting inferior performance on some music samples. Based on evaluation results, we conclude that under ultra-low-delay constraints, DualStream achieves reconstruction quality comparable to, or slightly lower than, the baseline NAC of similar sizes.

Finally, we conducted an AB preference test between Base_{32} and large $\mathrm{Dual}^{2:8}_{128:32}$ with $(C_{\mathrm{G}}, C_{\mathrm{L}}) = (52, 31)$, denoted as $\mathrm{Dual}^{2:8}_{128:32}$-L, to ensure similar computational demands. The results in Figure 5 shows that, under comparable computational budgets, DualStream outperforms the baseline NAC.

4.4. Analysis on Computational Complexity

Figure 6 illustrates the computational costs of each Encoder and Decoder Block from the baseline NACs and DualStream, measured in multiply-accumulate (MAC) operations per second of input audio. Computational costs of the baseline NACs drastically increase in the blocks near the bottleneck, as the overall down-sampling ratio decreases. This tendency is due to the increased input feature length in those layers, as the stride factor of the preceding down-sampling layer or the succeeding up-sampling layer decreases. In contrast, DualStream shows approximately 40% reduction in computational costs compared to the baseline NAC with a similar model size, as shown in Table 1, due to the computational efficiency of the Global Stream.

5. Conclusion

In this work, we propose DualStream, a novel NAC architecture that improves computational efficiency in extremely low-delay scenarios by integrating two NAC modules with small and large down-sampling ratios. Compared to conventional methods, which require high amounts of computation due to their small down-sampling ratios, DualStream benefits from a module with a large down-sampling ratio that is computationally efficient while imposing no additional latency. Experiments show that DualStream achieves reconstruction quality comparable to conventional NACs while maintaining a delay below 1 ms and reducing computational costs by approximately 40%.

6. References

[1] Y. Wang, "Low latency audio processing," Ph.D. dissertation, Queen Mary University of London, 2018.

[2] N. Kitawaki and K. Itoh, "Pure delay effects on speech quality in telecommunications," *IEEE Journal on selected Areas in Communications*, vol. 9, no. 4, pp. 586–593, 1991.

[3] J. Boley and M. Lester, "The effects of latency on live sound monitoring," in *Audio Engineering Society, Convention Paper*, 2007, p. 20.

[4] ISO/IEC 14496-3:2005, "Information technology - generic coding of moving pictures and associated audio information - part 3: Advanced audio coding (aac), subpart 4: General audio coding (ga) - aac, twinvq, bsac," 2005.

[5] ISO/IEC 14496-3:2005/FDAM9, "Information technology - generic coding of moving pictures and associated audio information - part 3: Advanced audio coding (aac), amendment 9: Enhanced low delay aac," 2007.

[6] J.-M. Valin, K. Vos, and T. Terriberry, "Definition of the opus audio codec," Tech. Rep., 2012.

[7] "SoundStream: An end-to-end neural audio codec," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 30, pp. 495–507, Nov. 2021.

[8] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *Transactions on Machine Learning Research*, 2023.

[9] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved rvqgan," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[10] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.

[11] A. Gritsenko, T. Salimans, R. van den Berg, J. Snoek, and N. Kalchbrenner, "A spectral energy distance for parallel speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 062–13 072, 2020.

[12] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in neural information processing systems*, vol. 33, pp. 17 022–17 033, 2020.

[13] J. Herre and S. Dick, "Psychoacoustic models for perceptual audio coding—a tutorial review," *Applied Sciences*, vol. 9, no. 14, p. 2854, 2019.

[14] K. Zhen, M. S. Lee, J. Sung, S. Beack, and M. Kim, "Psychoacoustic calibration of loss functions for efficient end-to-end neural audio coding," *IEEE Signal Processing Letters*, vol. 27, pp. 2159–2163, 2020.

[15] S. Shin, J. Byun, Y. Park, J. Sung, and S. Beack, "Deep neural network (dnn) audio coder using a perceptually improved training method," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 871–875.

[16] J. Byun, S. Shin, J. Sung, S. Beack, and Y. Park, "Optimization of deep neural network (dnn) speech coder using a multi time scale perceptual loss function." in *Interspeech*, 2022, pp. 4411–4415.

[17] M. Gayer, M. Lutzky, G. Schuller, U. Krämer, and S. Wabnik, "A guideline to audio codec delay," in *Audio Engineering Society Convention 116*. Audio Engineering Society, 2004.

[18] A. Vasuki and P. Vanathi, "A review of vector quantization techniques," *IEEE Potentials*, vol. 25, no. 4, pp. 39–47, 2006.

[19] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge, and Y. Wu, "Vector-quantized image modeling with improved vqgan," *International Conference on Learning Representations*, 2022.

[20] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[21] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *International Conference on Learning Representations*, 2016.

[22] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations*, 2018.

[23] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *International Conference on Learning Representations*, 2019.

[24] J. H. Lim and J. C. Ye, "Geometric gan," *arXiv preprint arXiv:1705.02894*, 2017.

[25] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[26] H. Dubey, V. Gopal, R. Cutler, S. Matusevych, S. Braun, E. S. Eskimez, M. Thakker, T. Yoshioka, H. Gamper, and R. Aichner, "Icassp 2022 deep noise suppression challenge," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 9271–9275.

[27] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "Musdb18-hq - an uncompressed version of musdb18," Aug. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3338373

[28] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The mtg-jamendo dataset for automatic music tagging," in *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*, Long Beach, CA, United States, 2019. [Online]. Available: http://hdl.handle.net/10230/42015

[29] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2022.

[30] M. Chinen, F. S. Lim, J. Skoglund, N. Gureev, F. O'Gorman, and A. Hines, "Visqol v3: An open source production ready objective speech and audio metric," in *2020 twelfth international conference on quality of multimedia experience (QoMEX)*. IEEE, 2020, pp. 1–6.

[31] T. Thiede, W. C. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J. G. Beerends, and C. Colomes, "Peaq-the itu standard for objective measurement of perceived audio quality," *Journal of the Audio Engineering Society*, vol. 48, no. 1/2, pp. 3–29, 2000.

[32] T. Kastner and J. Herre, "An efficient model for estimating subjective quality of separated audio source signals," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 95–99.

[33] G. J. Mysore, "Can we automatically transform speech recorded on common consumer devices in real-world environments into professional production quality speech?—a dataset, insights, and challenges," *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1006–1010, 2014.

[34] ITU-R Rec. BS.1534-1, "Method for the subjective assessment of intermediate quality level of coding systems," in *International Telecommunications Union, Geneva, Switzerland*, 2003.